Ruby master - Feature #16557
Deduplicate Regexp literals

01/23/2020 11:49 AM - byroot (Jean Boussier)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Open</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td></td>
</tr>
<tr>
<td>Target version:</td>
<td></td>
</tr>
</tbody>
</table>

**Description**

Pull Request: [https://github.com/ruby/ruby/pull/2859](https://github.com/ruby/ruby/pull/2859)

**Context**

Real world application contain many duplicated Regexp literals.

From a rails/console in Redmine:

```ruby
>> ObjectSpace.each_object(Regexp).count
=> 6828
>> ObjectSpace.each_object(Regexp).uniq.count
=> 4162
>> ObjectSpace.each_object(Regexp).to_a.map { |r| ObjectSpace.memssize_of(r) }.sum
=> 4611957 # 4.4 MB total
>> ObjectSpace.each_object(Regexp).to_a.map { |r| ObjectSpace.memssize_of(r) }.sum - ObjectSpace.each_object(Regexp).to_a.uniq.map { |r| ObjectSpace.memssize_of(r) }.sum
=> 1490601 # 1.42 MB could be saved
```

Here's the to 10 most duplicated regexps in Redmine:

147: /"/
107: /\s+/
103: /
89: /\n/
83: /'/
76: /\s+/m
37: /\d+/
35: /\[/
33: /./
33: /\./

Any empty Rails application will have a similar amount of regexps.

**The feature**

Since [https://bugs.ruby-lang.org/issues/16377](https://bugs.ruby-lang.org/issues/16377) made literal regexps frozen, it is possible to deduplicate literal regexps without changing any semantic and save a decent amount of resident memory.

**The patch**

I tried implementing this feature in a way very similar to the frozen_strings table, it's functional but I'm having trouble with a segfault on Linux: [https://github.com/ruby/ruby/pull/2859](https://github.com/ruby/ruby/pull/2859)

**History**

#1 - 01/23/2020 01:36 PM - Eregon (Benoit Daloze)
- Description updated

#2 - 01/23/2020 01:38 PM - Eregon (Benoit Daloze)
This is quite interesting, and would also avoid compiling these duplicated Regexp again, which likely saves quite a bit of startup time.

#3 - 01/23/2020 03:37 PM - byroot (Jean Boussier)
Eregon (Benoit Daloze) wrote:

would also avoid compiling these duplicated Regexp again.

In theory yes, however my current patch doesn't go that far for simplicity's sake. However that would indeed be a nice followup or improvement.